

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**





US Patent &amp; Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

reflective transparency + java

SEARCH

THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used **reflective transparency java**Found **4,976** of **141,345**

Sort results by

relevance

[Save results to a Binder](#)[Try an Advanced Search](#)

Display results

expanded form

[Search Tips](#)[Try this search in The ACM Guide](#)☐ Open results in a new window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐**1** [Reflections on remote reflection](#)

Michael Richmond, James Noble

January 2001 **Australian Computer Science Communications , Proceedings of the 24th Australasian conference on Computer science**, Volume 23 Issue 1

Full text available: pdf(939.96 KB)

[Publisher Site](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

The Java programming language provides both reflection and remote method invocation: reflection allows a program to inspect itself and its runtime environment, remote method invocation allows methods to be invoked transparently across a network. Unfortunately, the standard Java implementations of reflection and remote method invocation are incompatible: programmers cannot reflect on a remote application. We describe how Java systems can be extended to support **Remote Reflection** transparently ...

**2** [FlexiNet—a flexible component oriented middleware system](#)

Richard Hayton, Andrew Herbert, Douglas Donaldson

September 1998 **Proceedings of the 8th ACM SIGOPS European workshop on Support for composing distributed applications**

Full text available: pdf(1.20 MB)

Additional Information: [full citation](#), [citations](#), [index terms](#)**3** [Maya: multiple-dispatch syntax extension in Java](#)

Jason Baker, Wilson C. Hsieh

May 2002 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2002 Conference on Programming language design and implementation**, Volume 37 Issue 5

Full text available: pdf(152.75 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We have designed and implemented Maya, a version of Java that allows programmers to extend and reinterpret its syntax. Maya generalizes macro systems by treating grammar productions as generic functions, and semantic actions on productions as multimethods on the corresponding generic functions. Programmers can write new generic functions (i.e., grammar productions) and new multimethods (i.e., semantic actions), through which they can extend the grammar of the language and change the semantics of ...

**Keywords:** Java, generative programming, macros, metaprogramming



#### 4 Exploiting reflection in mobile computing middleware

Licia Capra, Gordon S. Blair, Cecilia Mascolo, Wolfgang Emmerich, Paul Grace

October 2002 **ACM SIGMOBILE Mobile Computing and Communications Review**, Volume 6 Issue 4

Full text available:  pdf(121.82 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The increasing popularity of portable devices and recent advances in wireless networking technologies facilitate the engineering of new classes of applications, which present challenging problems to designers. Mobile devices face temporary and unannounced loss of network connectivity when they are moved, they are likely to have scarce resources, and they are required to react to frequent changes in the environment. To accommodate these new requirements imposed by mobility, middleware platforms f ...

#### 5 System support for object groups

Rachid Guerraoui, Pascal Felber, Benoît Garbinato, Karim Mazouni

October 1998 **ACM SIGPLAN Notices , Proceedings of the 13th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**, Volume 33 Issue 10

Full text available:  pdf(2.12 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper draws several observations from our experiences in building support for object groups. These observations actually go beyond our experiences and may apply to many other developments of object based distributed systems. Our first experience aimed at building support for Smalltalk object replication using the Isis process group toolkit. It was quite easy to achieve group transparency but we were confronted with a strong mismatch between the rigidity of the process group model and the fle ...

#### 6 Technical papers: software architecture I: Design Pattern Rationale Graphs: linking design to source

Elisa L. A. Baniassad, Gail C. Murphy, Christa Schwanninger

May 2003 **Proceedings of the 25th international conference on Software engineering**

Full text available:  pdf(1.13 MB)  Additional Information: [full citation](#), [abstract](#), [references](#)  
[Publisher Site](#)

A developer attempting to evolve a system in which design patterns have been applied can benefit from knowing which code implements which design pattern. For instance, the developer may be able to understand the purpose, or to assess the flexibility of the code, more quickly. The degree to which the developer benefits depends upon their understanding of the pattern. Achieving an in-depth understanding of even a simple pattern can be difficult as pattern descriptions span several pages of text, a ...

#### 7 The virtual reality modeling language and Java

Don Brutzman

June 1998 **Communications of the ACM**, Volume 41 Issue 6

Full text available:  pdf(763.87 KB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

#### 8 Embedded hardware design case studies: Design flow for HW / SW acceleration transparency in the thumbpod secure embedded system

David Hwang, Bo-Cheng Lai, Patrick Schaumont, Kazuo Sakiyama, Yi Fan, Shenglin Yang, Alireza Hodjat, Ingrid Verbauwhede

June 2003 **Proceedings of the 40th conference on Design automation**



Full text available:  pdf(250.69 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper describes a case study and design flow of a secure embedded system called ThumbPod, which uses cryptographic and biometric signal processing acceleration. It presents the concept of HW/SW acceleration transparency, a systematic method to accelerate Java functions in both software and hardware. An example of acceleration transparency for a Rijndael encryption function is presented. The embedded prototype hardware platform is also described. Acceleration transparency yields software and ...

## 9 [Interactive posters: Kits for learning and a kit for kitmaking](#)

Carol Strohecker, Adrienne Slaughter

April 2000 **CHI '00 extended abstracts on Human factors in computing systems**

Full text available:  pdf(217.26 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

We bring together concerns in software design and learning theory through creation of a Java framework for development of software construction kits. The kits are highly visual and highly interactive, and are premised on the notion of "microworlds" as environments for learning and learning research [6]. Usage of four existing kits is informing development of the framework, which in turn we are applying to development of a new kit. The kits support construction of two-dimensional, graphical struc ...

**Keywords:** Java framework, design patterns, learning, microworlds

## 10 [Coordination models, languages and applications \(CM\): Enforcing agent communication laws by means of a reflective framework](#)

Antonella Di Stefano, Corrado Santoro, Giuseppe Pappalardo, Emiliano Tramontana

March 2004 **Proceedings of the 2004 ACM symposium on Applied computing**

Full text available:  pdf(155.54 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Agent Coordination Contexts (ACCs) have been proposed as virtual environments where agents live and interact. In this way, as in a human society, interactions may be subjected to conventions and laws depending on their context. This is obtained by a suitable ACC that embeds the communication laws relevant to a specific application and checks whether they are fulfilled as interactions take place. Context modelling, while representing a communication aspect relevant for all the agents of an applica ...

**Keywords:** agents, communication, computational reflection, software engineering

## 11 [Programming languages for mobile code](#)

Tommy Thorn

September 1997 **ACM Computing Surveys (CSUR)**, Volume 29 Issue 3

Full text available:  pdf(393.65 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Sun's announcement of the programming language Java more than anything popularized the notion of mobile code, that is, programs traveling on a heterogeneous network and automatically executing upon arrival at the destination. We describe several classes of mobile code and extract their common characteristics, where security proves to be one of the major concerns. With these characteristics as reference points, we examine six representative languages proposed for mobile code. The conclusion ...

**Keywords:** Java, Limbo, Objective Caml, Obliq, Safe-Tcl, distribution, formal methods, mobile code, network programming, object orientation, portability, safety, security, telescript



12 Flexible collaboration transparency: supporting worker independence in replicated application-sharing systems

James Begole, Mary Beth Rosson, Clifford A. Shaffer

June 1999 **ACM Transactions on Computer-Human Interaction (TOCHI)**, Volume 6 Issue 2

Full text available:  pdf(312.22 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

This article presents a critique of conventional collaboration transparency systems, also called "application-sharing" systems, which provide the real-time shared use of legacy single-user applications. We find that conventional collaboration transparency systems are inefficient in their use of network resources and lack support for key groupware principles: concurrent work, relaxed WYSIWIS, and group awareness. Next, we present an alternative approach to implementing collaborat ...

**Keywords:** Flexible JAMM, Java, application sharing, collaboration transparency, computer-supported cooperative work, groupware, usability

13 Partial behavioral reflection: spatial and temporal selection of reification

Éric Tanter, Jacques Noyé, Denis Caromel, Pierre Cointe

October 2003 **ACM SIGPLAN Notices , Proceedings of the 18th ACM SIGPLAN conference on Object-oriented programing, systems, languages, and applications**, Volume 38 Issue 11

Full text available:  pdf(261.44 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Behavioral reflection is a powerful approach for adapting the behavior of running applications. In this paper we present and motivate *partial behavioral reflection*, an approach to more efficient and flexible behavioral reflection. We expose the *spatial* and *temporal* dimensions of such reflection, and propose a model of partial behavioral reflection based on the notion of *hooksets*. In the context of Java, we describe a reflective architecture offering appropriate interf ...

**Keywords:** aspect-oriented programming, open systems, reflection

14 Java based conservative distributed simulation

Alois Ferscha, Michael Richter

December 1997 **Proceedings of the 29th conference on Winter simulation**

Full text available:  pdf(835.03 KB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

15 Turning light bulbs into objects

Bernd Bruegge, Truman Fenton, Tae Wook Kim, Ricardo Pravia, Aseem Sharma, Benedict Fernandes, Seongju Chang, Volker Hartkopf

January 1997 **Addendum to the 1997 ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications (Addendum)**

Full text available:  pdf(683.50 KB)

Additional Information: [full citation](#), [references](#), [index terms](#)

16 Configuration and flexibility: Reifying communication at the application level

Andrew P. Black, Jie Huang, Jonathan Walpole

October 2001 **Proceedings of the 2001 international workshop on Multimedia middleware**



Full text available:  pdf(473.47 KB) Additional Information: [full citation](#), [abstract](#), [references](#)

We describe an abstraction for reifying communication in information flow systems. Thus, our goal is not to hide communication, but to represent it explicitly as objects that the program can interrogate and manipulate. Moreover, these objects represent communication in application-level terms, not in terms of network or process implementation.

**Keywords:** InfoPipes, MIDI, communication middleware, information flow, infosphere, reification, transparency

## 17 Collaboration transparency in the DISCIPLE framework

Wen Li, Weicong Wang, Ivan Marsic

November 1999 **Proceedings of the international ACM SIGGROUP conference on Supporting group work**

Full text available:  pdf(2.04 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Sharing single-user software applications is a major goal of synchronous groupware particularly because the majority of applications continues to be developed for single users. We present a mechanism for sharing collaboration-transparent single-user applications in our DISCIPLE collaboration framework. DISCIPLE is the equivalent of a Web browser that allows sharing applets (Java components, both transparent and aware of collaboration). It allows users with no programming background to quick ...

**Keywords:** CSCW frameworks, JavaBeans, collaboration-transparent applications, synchronous groupware

## 18 Full papers: Explicit programming

Avi Bryant, Andrew Catton, Kris De Volder, Gail C. Murphy

April 2002 **Proceedings of the 1st international conference on Aspect-oriented software development**

Full text available:  pdf(856.41 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citings](#), [index terms](#)

Many design concepts can be expressed only indirectly in source code. When this occurs, a single concept at design results in a verbose amount of code that is scattered across the system structure. In this paper, we present explicit programming, an approach that enables a developer to introduce new vocabulary into the source to capture a design concept explicitly. An introduced vocabulary item modularizes the implementation details associated with a design concept, reducing the scattering of code ...

## 19 Fast detection of communication patterns in distributed executions

Thomas Kunz, Michiel F. H. Seuren

November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research**

Full text available:  pdf(4.21 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

## 20 Programming languages (PL): Implementing the essence of reflection: a reflective run-time environment



Massimo Ancona, Walter Cazzola

March 2004 **Proceedings of the 2004 ACM symposium on Applied computing**

Full text available:  pdf(260.53 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Computational reflection provides the developers with a programming mechanism devoted to favorite code extensibility, reuse and maintenance. Notwithstanding that, it has not achieved developers' unanimous acceptance and its full potential yet. In our opinion, this depends on the intrinsic complexity of most of the reflective approaches that hinders their efficient implementation. The aim of this paper consists of defining the essence of reflection, that is, to identify the minimal set of charact ...

**Keywords:** compiler construction, reflection, run-time environment

Results 1 - 20 of 200

Result page: **1** [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)





US Patent &amp; Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

reflective transparency + java



THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used **reflective transparency java**Found **4,976** of **141,345**

Sort results by

relevance

Display results

expanded form

[Save results to a Binder](#)[Search Tips](#)[Open results in a new window](#)[Try an Advanced Search](#)[Try this search in The ACM Guide](#)

Results 21 - 40 of 200

Result page: [previous](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐**21 Software engineering and middleware: a roadmap**

Wolfgang Emmerich

May 2000 **Proceedings of the conference on The future of Software engineering**Full text available: pdf(1.34 MB) Additional Information: [full citation](#), [references](#), [citings](#), [index terms](#)**22 AdJava: automatic distribution of Java applications**

Mohammad M. Fuad, Michael J. Oudshoorn

January 2002 **Australian Computer Science Communications , Proceedings of the twenty-fifth Australasian conference on Computer science - Volume 4,**  
Volume 24 Issue 1Full text available: pdf(1.27 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The majority of the world's computing resources remains idle most of the time. By using this resource pool, an individual computation may be completed in a fraction of time required to run the same computation on a single machine. However, distributing a program over a number of machines proves to be a tedious and difficult job. This paper introduces a system, called AdJava, which harnesses the computing power of these under-utilized heterogeneous computers by automatically distributing the user ...

**Keywords:** distributed programming, software agents.**23 Tutorials: Component technologies: Java beans, COM, CORBA, RMI, EJB and the CORBA component model**

Wolfgang Emmerich, Nima Kaveh



May 2002 **Proceedings of the 24th international conference on Software engineering**Full text available: pdf(220.94 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citings](#), [index terms](#)

This one-day tutorial is aimed at software engineering practitioners and researchers, who are familiar with object-oriented analysis, design and programming and want to obtain an overview of the technologies that are enabling component-based development. We introduce the idea of component-based development by defining the concept and providing its economic rationale. We describe how object-oriented programming evolved into local component models, such as Java Beans and distributed object technol ...



**24** StratOSphere: mobile processing of distributed objects in Java

Daniel Wu, Divyakant Agrawal, Amr El Abbadi

October 1998 **Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking**Full text available:  pdf(1.38 MB)Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)**25** Process migrationSeptember 2000 **ACM Computing Surveys (CSUR)**, Volume 32 Issue 3Full text available:  pdf(1.24 MB)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Process migration is the act of transferring a process between two machines. It enables dynamic load distribution, fault resilience, eased system administration, and data access locality. Despite these goals and ongoing research efforts, migration has not achieved widespread use. With the increasing deployment of distributed systems in general, and distributed operating systems in particular, process migration is again receiving more attention in both research and product development. As hi ...


**Keywords:** distributed operating systems, distributed systems, load distribution, process migration

**26** Report on the eighth ACM SIGOPS European workshop

Jean Bacon

January 1999 **ACM SIGOPS Operating Systems Review**, Volume 33 Issue 1Full text available:  pdf(988.38 KB)Additional Information: [full citation](#), [index terms](#)**27** Adaptive middleware: The case for reflective middleware

Fabio Kon, Fabio Costa, Gordon Blair, Roy H. Campbell

June 2002 **Communications of the ACM**, Volume 45 Issue 6Full text available:  pdf(114.33 KB)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#) html(32.07 KB)

It's flexible and reconfigurable yet simple for programmers to use, notably for building dynamic distributed applications operating on the Net.

**28** Design pattern implementation in Java and aspectJ

Jan Hannemann, Gregor Kiczales

November 2002 **ACM SIGPLAN Notices , Proceedings of the 17th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**, Volume 37 Issue 11Full text available:  pdf(366.95 KB)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

AspectJ implementations of the GoF design patterns show modularity improvements in 17 of 23 cases. These improvements are manifested in terms of better code locality, reusability, composability, and (un)pluggability. The degree of improvement in implementation modularity varies, with the greatest improvement coming when the pattern solution structure involves crosscutting of some form, including one object playing multiple roles, many objects playing one role, or an object playing roles in multip ...

**Keywords:** aspect-oriented programming, design patterns



29 A comparison of three approaches to language, compiler, and library support for multidimensional arrays in Java

José E. Moreira, Samuel P. Midkiff, Manish Gupta

June 2001 **Proceedings of the 2001 joint ACM-ISCOPE conference on Java Grande**

Full text available:  pdf(790.28 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The lack of direct support for multidimensional arrays in Java™ has been recognized as a major deficiency in the language's applicability to numerical computing. The typical approach to adding multidimensional arrays to Java has been through class libraries that implement these structures. It has been shown that the class library approach can achieve very high-performance for numerical computing, through the use of compiler techniques and efficient implementations of aggregate array operations ...

30 Dynamic translation: Jumbo: run-time code generation for Java and its applications

Sam Kamin, Lars Clausen, Ava Jarvis

March 2003 **Proceedings of the international symposium on Code generation and optimization: feedback-directed and runtime optimization**

Full text available:  pdf(698.19 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)  
 [Publisher Site](#)

Run-time code generation is a well-known technique for improving the efficiency of programs by exploiting dynamic information. Unfortunately, the difficulty of constructing run-time code-generators has hampered their widespread use. We describe Jumbo, a tool for easily creating run-time code generators for Java. Jumbo is a compiler for a two-level version of Java, where programs can contain quoted code fragments. The Jumbo API allows the code fragments to be combined at run-time and then execute ...

**Keywords:** Java, run-time code generation

31 Middleware for dependable network services in partitionable distributed systems

Alberto Montresor, Renzo Davoli, Özalp Babaoğlu

January 2001 **ACM SIGOPS Operating Systems Review**, Volume 35 Issue 1

Full text available:  pdf(1.38 MB) Additional Information: [full citation](#), [abstract](#), [index terms](#)

We describe the design and implementation of *Jgroup*: a middleware system that integrates group technology with distributed objects and is based on Java RMI. Jgroup supports a programming paradigm called *object groups* and enables development of dependable network services based on replication. Among the novel features of Jgroup is a uniform object-oriented interface for programming both services and their clients. The fact that Jgroup exposes network effects, including partitions, t ...

32 Session B: Computer graphics in education: Experiences in porting a virtual reality system to Java

Shaun Bangay

November 2001 **Proceedings of the 1st international conference on Computer graphics, virtual reality and visualisation**

Full text available:  pdf(527.43 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Practical experience in porting a large virtual reality system from C/C++ to Java indicates that porting this type of real-time application is both feasible, and has several merits. The ability to transfer objects in space and time allows useful facilities such as distributed agent support and persistence to be added. Reflection and type comparisons allow flexible manipulations of objects of different types at run-time. Native calls and native code compilation reduce or remove the overhead of in ...



**Keywords:** Java, native calls, networking, serialization

33 Dynamic layout of distributed applications in FarGo

Ophir Holder, Israel Ben-Shaul, Hovav Gazit

May 1999 **Proceedings of the 21st international conference on Software engineering**


Full text available:  pdf(1.45 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

**Keywords:** Java, distributed components, dynamic objects, engineering distributed systems, mobile objects

34 Type-Safe linking with recursive DLLs and shared libraries

Dominic Duggan

November 2002 **ACM Transactions on Programming Languages and Systems (TOPLAS)**,  
Volume 24 Issue 6

Full text available:  pdf(658.62 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Component-based programming is an increasingly prevalent theme in software development, motivating the need for expressive and safe module interconnection languages. Dynamic linking is an important requirement for module interconnection languages, as exemplified by dynamic link libraries (DLLs) and class loaders in operating systems and Java, respectively. A semantics is given for a type-safe module interconnection language that supports shared libraries and dynamic linking, as well as circular ...

**Keywords:** Dynamic Linking, Module Interconnection Languages, Recursive Modules, Shared Libraries

35 Jazz: an extensible zoomable user interface graphics toolkit in Java

Benjamin B. Bederson, Jon Meyer, Lance Good

November 2000 **Proceedings of the 13th annual ACM symposium on User interface software and technology**

Full text available:  pdf(137.37 KB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

**Keywords:** Jazz, Pad++, animation, graphics, user interface management systems (UIMS), zoomable user interfaces (ZUIs)

36 Wire-area parallel computing in Java

Rob van Nieuwpoort, Jason Maassen, Henri E. Bal, Thilo Kielmann, Ronald Veldema

June 1999 **Proceedings of the ACM 1999 conference on Java Grande**

Full text available:  pdf(839.20 KB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

37 On type systems for object-oriented database programming languages

Yuri Leontiev, M. Tamer Özsu, Duane Szafron

December 2002 **ACM Computing Surveys (CSUR)**, Volume 34 Issue 4

Full text available:  pdf(346.87 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)



The concept of an object-oriented database programming language (OODBPL) is appealing because it has the potential of combining the advantages of object orientation and database programming to yield a powerful and universal programming language design. A uniform and consistent combination of object orientation and database programming, however, is not straightforward. Since one of the main components of an object-oriented programming language is its type system, one of the first problems that ar ...

**Keywords:** OODB, OODBPL, object-oriented database programming language, type checking, typing

38 Web-based and Java-based simulation: A review of web based simulation: whither we wander?

Jasna Kuljis, Ray J. Paul

December 2000 **Proceedings of the 32nd conference on Winter simulation**

Full text available:  pdf(226.42 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citing](#)

This paper considers a variety of new technologies for discrete-event simulation software development. Environments and languages for web based simulation are reviewed. Web based applications are discussed. After proposing a summary of the review, ways of working that will have an unpredictable effect on the future of simulation modeling are proposed.

39 Application isolation in the Java Virtual Machine

Grzegorz Czajkowski

October 2000 **ACM SIGPLAN Notices , Proceedings of the 15th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**, Volume 35 Issue 10

Full text available:  pdf(217.49 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citing](#), [index terms](#)

To date, systems offering multitasking for the Java<sup>®</sup>; programming language either use one process or one class loader for each application. Both approaches are unsatisfactory. Using operating system processes is expensive, scales poorly and does not fully exploit the protection features inherent in a safe language. Class loaders replicate application code, obscure the type system, and non-uniformly treat 'trusted' and 'untrusted' classes, which leads to subtle, but nevertheless, potenti ...

**Keywords:** Java Virtual Machine, application isolation, multitasking

40 Technical papers: NETKIT: a software component-based approach to programmable networking

Geoff Coulson, Gordon Blair, David Hutchison, Ackbar Joolia, Kevin Lee, Jo Ueyama, Antonio Gomes, Yimin Ye

October 2003 **ACM SIGCOMM Computer Communication Review**, Volume 33 Issue 5

Full text available:  pdf(316.64 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

While there has already been significant research in support of openness and programmability in networks, this paper argues that there remains a need for generic support for the integrated development, deployment and management of programmable networking software. We further argue that this support should explicitly address the management of run-time reconfiguration of systems, and should be independent of any particular programming paradigm (e.g. active networking or open signaling), programmin ...

**Keywords:** components, middleware, programmable networking, reflection



Results 21 - 40 of 200

Result page: [previous](#) [1](#) **[2](#)** [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)





US Patent &amp; Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

reflective transparency + java

SEARCH

THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used **reflective transparency java**Found **4,976** of **141,345**

Sort results by

relevance

[Save results to a Binder](#)[Try an Advanced Search](#)

Display results

expanded form

[Search Tips](#)[Try this search in The ACM Guide](#)
☐ Open results in a new window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐**1** [Reflections on remote reflection](#)

Michael Richmond, James Noble

 January 2001 **Australian Computer Science Communications , Proceedings of the 24th Australasian conference on Computer science**, Volume 23 Issue 1

Full text available: pdf(939.96 KB)

[Publisher Site](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

The Java programming language provides both reflection and remote method invocation: reflection allows a program to inspect itself and its runtime environment, remote method invocation allows methods to be invoked transparently across a network. Unfortunately, the standard Java implementations of reflection and remote method invocation are incompatible: programmers cannot reflect on a remote application. We describe how Java systems can be extended to support **Remote Reflection** transparently ...

**2** [FlexiNet—a flexible component oriented middleware system](#)

Richard Hayton, Andrew Herbert, Douglas Donaldson

 September 1998 **Proceedings of the 8th ACM SIGOPS European workshop on Support for composing distributed applications**

Full text available: pdf(1.20 MB)

Additional Information: [full citation](#), [citations](#), [index terms](#)**3** [Maya: multiple-dispatch syntax extension in Java](#)

Jason Baker, Wilson C. Hsieh

 May 2002 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2002 Conference on Programming language design and implementation**, Volume 37 Issue 5

Full text available: pdf(152.75 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We have designed and implemented Maya, a version of Java that allows programmers to extend and reinterpret its syntax. Maya generalizes macro systems by treating grammar productions as generic functions, and semantic actions on productions as multimethods on the corresponding generic functions. Programmers can write new generic functions (i.e., grammar productions) and new multimethods (i.e., semantic actions), through which they can extend the grammar of the language and change the semantics of ...

**Keywords:** Java, generative programming, macros, metaprogramming



#### 4 Exploiting reflection in mobile computing middleware

Licia Capra, Gordon S. Blair, Cecilia Mascolo, Wolfgang Emmerich, Paul Grace

October 2002 **ACM SIGMOBILE Mobile Computing and Communications Review**, Volume 6 Issue 4


Full text available:  pdf(121.82 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The increasing popularity of portable devices and recent advances in wireless networking technologies facilitate the engineering of new classes of applications, which present challenging problems to designers. Mobile devices face temporary and unannounced loss of network connectivity when they are moved, they are likely to have scarce resources, and they are required to react to frequent changes in the environment. To accommodate these new requirements imposed by mobility, middleware platforms f ...

#### 5 System support for object groups

Rachid Guerraoui, Pascal Felber, Benoît Garbinato, Karim Mazouni

October 1998 **ACM SIGPLAN Notices , Proceedings of the 13th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**, Volume 33 Issue 10

Full text available:  pdf(2.12 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper draws several observations from our experiences in building support for object groups. These observations actually go beyond our experiences and may apply to many other developments of object based distributed systems. Our first experience aimed at building support for Smalltalk object replication using the Isis process group toolkit. It was quite easy to achieve group transparency but we were confronted with a strong mismatch between the rigidity of the process group model and the fle ...

#### 6 Technical papers: software architecture I: Design Pattern Rationale Graphs: linking design to source

Elisa L. A. Baniassad, Gail C. Murphy, Christa Schwanninger

May 2003 **Proceedings of the 25th international conference on Software engineering**

Full text available:  pdf(1.13 MB)  Additional Information: [full citation](#), [abstract](#), [references](#)  
[Publisher Site](#)

A developer attempting to evolve a system in which design patterns have been applied can benefit from knowing which code implements which design pattern. For instance, the developer may be able to understand the purpose, or to assess the flexibility of the code, more quickly. The degree to which the developer benefits depends upon their understanding of the pattern. Achieving an in-depth understanding of even a simple pattern can be difficult as pattern descriptions span several pages of text, a ...

#### 7 The virtual reality modeling language and Java

Don Brutzman

June 1998 **Communications of the ACM**, Volume 41 Issue 6

Full text available:  pdf(763.87 KB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

#### 8 Embedded hardware design case studies: Design flow for HW / SW acceleration transparency in the thumbpod secure embedded system

David Hwang, Bo-Cheng Lai, Patrick Schaumont, Kazuo Sakiyama, Yi Fan, Shenglin Yang, Alireza Hodjat, Ingrid Verbauwhede

June 2003 **Proceedings of the 40th conference on Design automation**



Full text available:  pdf(250.69 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper describes a case study and design flow of a secure embedded system called ThumbPod, which uses cryptographic and biometric signal processing acceleration. It presents the concept of HW/SW acceleration transparency, a systematic method to accelerate Java functions in both software and hardware. An example of acceleration transparency for a Rijndael encryption function is presented. The embedded prototype hardware platform is also described. Acceleration transparency yields software and ...

## 9 [Interactive posters: Kits for learning and a kit for kitmaking](#)

Carol Strohecker, Adrienne Slaughter

April 2000 **CHI '00 extended abstracts on Human factors in computing systems**

Full text available:  pdf(217.26 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

We bring together concerns in software design and learning theory through creation of a Java framework for development of software construction kits. The kits are highly visual and highly interactive, and are premised on the notion of "microworlds" as environments for learning and learning research [6]. Usage of four existing kits is informing development of the framework, which in turn we are applying to development of a new kit. The kits support construction of two-dimensional, graphical struc ...

**Keywords:** Java framework, design patterns, learning, microworlds

## 10 [Coordination models, languages and applications \(CM\): Enforcing agent communication laws by means of a reflective framework](#)

Antonella Di Stefano, Corrado Santoro, Giuseppe Pappalardo, Emiliano Tramontana

March 2004 **Proceedings of the 2004 ACM symposium on Applied computing**

Full text available:  pdf(155.54 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Agent Coordination Contexts (ACCs) have been proposed as virtual environments where agents live and interact. In this way, as in a human society, interactions may be subjected to conventions and laws depending on their context. This is obtained by a suitable ACC that embeds the communication laws relevant to a specific application and checks whether they are fulfilled as interactions take place. Context modelling, while representing a communication aspect relevant for all the agents of an applica ...

**Keywords:** agents, communication, computational reflection, software engineering

## 11 [Programming languages for mobile code](#)

Tommy Thorn

September 1997 **ACM Computing Surveys (CSUR)**, Volume 29 Issue 3

Full text available:  pdf(393.65 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Sun's announcement of the programming language Java more than anything popularized the notion of mobile code, that is, programs traveling on a heterogeneous network and automatically executing upon arrival at the destination. We describe several classes of mobile code and extract their common characteristics, where security proves to be one of the major concerns. With these characteristics as reference points, we examine six representative languages proposed for mobile code. The conclusion ...

**Keywords:** Java, Limbo, Objective Caml, Obliq, Safe-Tcl, distribution, formal methods, mobile code, network programming, object orientation, portability, safety, security, telescript



12 Flexible collaboration transparency: supporting worker independence in replicated application-sharing systems

James Begole, Mary Beth Rosson, Clifford A. Shaffer

June 1999 **ACM Transactions on Computer-Human Interaction (TOCHI)**, Volume 6 Issue 2

Full text available:  pdf(312.22 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

This article presents a critique of conventional collaboration transparency systems, also called "application-sharing" systems, which provide the real-time shared use of legacy single-user applications. We find that conventional collaboration transparency systems are inefficient in their use of network resources and lack support for key groupware principles: concurrent work, relaxed WYSIWIS, and group awareness. Next, we present an alternative approach to implementing collaborat ...

**Keywords:** Flexible JAMM, Java, application sharing, collaboration transparency, computer-supported cooperative work, groupware, usability

13 Partial behavioral reflection: spatial and temporal selection of reification

Éric Tanter, Jacques Noyé, Denis Caromel, Pierre Cointe

October 2003 **ACM SIGPLAN Notices , Proceedings of the 18th ACM SIGPLAN conference on Object-oriented programing, systems, languages, and applications**, Volume 38 Issue 11

Full text available:  pdf(261.44 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Behavioral reflection is a powerful approach for adapting the behavior of running applications. In this paper we present and motivate *partial behavioral reflection*, an approach to more efficient and flexible behavioral reflection. We expose the *spatial* and *temporal* dimensions of such reflection, and propose a model of partial behavioral reflection based on the notion of *hooksets*. In the context of Java, we describe a reflective architecture offering appropriate interf ...

**Keywords:** aspect-oriented programming, open systems, reflection

14 Java based conservative distributed simulation

Alois Ferscha, Michael Richter

December 1997 **Proceedings of the 29th conference on Winter simulation**

Full text available:  pdf(835.03 KB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

15 Turning light bulbs into objects

Bernd Bruegge, Truman Fenton, Tae Wook Kim, Ricardo Pravia, Aseem Sharma, Benedict Fernandes, Seongju Chang, Volker Hartkopf

January 1997 **Addendum to the 1997 ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications (Addendum)**

Full text available:  pdf(683.50 KB) Additional Information: [full citation](#), [references](#), [index terms](#)

16 Configuration and flexibility: Reifying communication at the application level

Andrew P. Black, Jie Huang, Jonathan Walpole

October 2001 **Proceedings of the 2001 international workshop on Multimedia middleware**



Full text available:  pdf(473.47 KB) Additional Information: [full citation](#), [abstract](#), [references](#)

We describe an abstraction for reifying communication in information flow systems. Thus, our goal is not to hide communication, but to represent it explicitly as objects that the program can interrogate and manipulate. Moreover, these objects represent communication in application-level terms, not in terms of network or process implementation.

**Keywords:** InfoPipes, MIDI, communication middleware, information flow, infosphere, reification, transparency

17 Collaboration transparency in the DISCIPLE framework

Wen Li, Weicong Wang, Ivan Marsic

November 1999 **Proceedings of the international ACM SIGGROUP conference on Supporting group work**

Full text available:  pdf(2.04 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Sharing single-user software applications is a major goal of synchronous groupware particularly because the majority of applications continues to be developed for single users. We present a mechanism for sharing collaboration-transparent single-user applications in our DISCIPLE collaboration framework. DISCIPLE is the equivalent of a Web browser that allows sharing applets (Java components, both transparent and aware of collaboration). It allows users with no programming background to quick ...

**Keywords:** CSCW frameworks, JavaBeans, collaboration-transparent applications, synchronous groupware

18 Full papers: Explicit programming

Avi Bryant, Andrew Catton, Kris De Volder, Gail C. Murphy

April 2002 **Proceedings of the 1st international conference on Aspect-oriented software development**

Full text available:  pdf(856.41 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Many design concepts can be expressed only indirectly in source code. When this occurs, a single concept at design results in a verbose amount of code that is scattered across the system structure. In this paper, we present explicit programming, an approach that enables a developer to introduce new vocabulary into the source to capture a design concept explicitly. An introduced vocabulary item modularizes the implementation details associated with a design concept, reducing the scattering of code ...

19 Fast detection of communication patterns in distributed executions

Thomas Kunz, Michiel F. H. Seuren

November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research**

Full text available:  pdf(4.21 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

20 Programming languages (PL): Implementing the essence of reflection: a reflective run-time environment



Massimo Ancona, Walter Cazzola

March 2004 **Proceedings of the 2004 ACM symposium on Applied computing**

Full text available:  pdf(260.53 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Computational reflection provides the developers with a programming mechanism devoted to favorite code extensibility, reuse and maintenance. Notwithstanding that, it has not achieved developers' unanimous acceptance and its full potential yet. In our opinion, this depends on the intrinsic complexity of most of the reflective approaches that hinders their efficient implementation. The aim of this paper consists of defining the essence of reflection, that is, to identify the minimal set of charact ...

**Keywords:** compiler construction, reflection, run-time environment

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)